Sparse data formats and efficient numerical methods for uncertainties quantification in numerical aerodynamics

A. Litvinenko
H. G. Matthies

Braunschweig : Institut für Wissenschaftliches Rechnen, 2010

(Informatik-Bericht 2010-01)

Veröffentlicht: 17.11.2010

http://www.digibib.tu-bs.de/?docid=00036490

Sparse data formats and efficient numerical methods for uncertainties quantification in numerical aerodynamics

A. Litvinenko and H. G. Matthies
wire@tu-bs.de
Institut für Wissenschaftliches Rechnen, Braunschweig, Germany

Abstract

The problem to be considered is the stationar system of Navier-Stokes equations with uncertain parameters and uncertain computational domain. We research how uncertainties in the angle of attack, in the Mach number and in the geometry of the airfoil propagate in the solution. The uncertain solution of this problem (pressure, density, velocity etc) is approximated via random fields. Since the whole set of realisations of these random fields are too much information, we demonstrate an algorithm of their low-rank approximation. This algorithm, working on the fly, is based on the QR-decomposition and has a linear complexity. This low-rank approximation allows us an effective postprocessing (computation of the mean value, variance, exceedance probability) with drastically reduced memory requirements.

Keywords: uncertainty quantification, stochastic Navier-Stokes, Karhunen-Loève expansion, polynomial chaos expansion, QR-algorithm, low-rank data format.

1 Introduction

Nowadays the trend of numerical mathematics is often trying to resolve inexact mathematical models by very exact deterministic numerical methods. The reason of this inexactness is that almost each mathematical model of a real world situation contains uncertainties in the coefficients, right-hand side, boundary conditions, initial data as well as in the computational geometry. All these uncertainties can affect the solution dramatically, which is, in its

Correspondence: A. Litvinenko, Institut für Wissenschaftliches Rechnen, Hans-Sommer Str. 65, 38106, Braunschweig, Germany

turn, also uncertain. In such a case the information of the interest is not the whole set of realisations of the solutions (too much data), but cumulative distribution function, density function, mean value, variance, exceedance probability etc.

During the last few years one can see an increasing interest in numerical methods for solving stochastic computational fluid dynamic (CFD) problems [2, 5, 13, 16, 20, 22]. In this work we consider a problem from aerodynamic, described by a system of Navier-Stokes equations, where uncertainties are modelled via random variables and random fields [11, 10]. We assume that there is a solver which is able to solve the deterministic Navier-Stokes problem. We also assume that spatial discretisation of the airfoil and the fluid around it is given. Our job is appropriate modelling of uncertainties, discretisation of resulting stochastic operator and developing stochastic/statistical numerical techniques for further quantification of uncertainties. At the same time, due to high complexity of the deterministic solver, we are interesting only in non-intrusive stochastic methods such as Monte Carlo or collocation methods. So, we are interesting in methods which do not require changes in the deterministic code.

The rest of the paper is structured as follows. In Section 2 we describe discretisation of random fields. For this purpose we apply the truncated Karhunen-Loève expansion (KLE) [12] and polynomial chaos expansion (PCE) of Wiener [21]. In Section 3 we explain how we model uncertainties in the parameters angle of attack and Mach number, in the atmosphere (Section 3.2) and uncertainties in the geometry (Section 3.3). To avoid large memory requirements and to reduce computing time, data sparse techniques for representation of input and output data (solution) were developed in Section 4. The whole set of realisations of the solution is compressed via the algorithm based on the singular value decomposition. The short idea is as follows.
Let $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1..Z$, stochastic realisations of the solution (already centred). For a small Z we build from all vectors $\mathbf{v}_i$ the matrix $W := [\mathbf{v}_1, ..., \mathbf{v}_Z] \in \mathbb{R}^{n \times Z}$ and compute its low-rank approximation $\tilde{W} = AB^T$, where $A \in \mathbb{R}^{n \times k}$ and $B \in \mathbb{R}^{Z \times k}$. For every new vector $\mathbf{v}_{Z+1}$ an update for the matrices A and B is computed on the fly with a linear complexity.
Section 5 is devoted to the numerical results, where we demonstrate the influence of uncertainties in the angle of attack α, in the Mach number Ma and in the airfoil geometry on the solution - drag, lift, pressure and absolute friction coefficients. The strongly reduced memory requirement for storage stochastic realisations of the solution is demonstrated as well.

2 Discretisation techniques

In the following, $(\Omega, \mathcal{B}, \mathbb{P})$ denotes a probability space, where Ω is the set of elementary events, $\mathcal{B}$ is the σ-algebra of events and $\mathbb{P}$ is the probability measure. The symbol ω always specifies an elementary event $\omega \in \Omega$. The problem to be consider in this work is the stationary system of Navier-Stokes equations with uncertain coefficients and parameters:

$$\begin{aligned} &v(x,\omega)\cdot\nabla v(x,\omega) - \tfrac{1}{Re}\nabla^2 v(x,\omega) + \nabla p(x,\omega) = g(x,\omega) \quad x \in \mathcal{G},\ \omega \in \Omega \\ &\nabla \cdot v(x,\omega) = 0 \end{aligned} \tag{1}$$

with some initial and boundary conditions. Here v is velocity, p pressure and g the right-hand side, the computational domain is RAE-2822 airfoil with some area around. Examples of uncertain parameters are the angle of attack α and the Mach number Ma. Uncertainties in the airfoil geometry are modelled via random field $\kappa(x,\omega)$ (see Section 3.3). For the numerical solution of (1) the presented input and output random fields need to be discretised both in the stochastic and in the spatial dimensions. One of the main tools here is the Karhunen-Loève expansion (KLE) [12]. Thus, an effective and "sparse" computation of the KLE is one of key points in solving Eq. 1. By definition, the Karhunen-Loève expansion (KLE) of a random field $\kappa(x,\omega)$ is the following series [12]

$$\kappa(x,\omega) = E_\kappa(x) + \sum_{\ell=1}^{\infty} \sqrt{\lambda_\ell}\phi_\ell(x)\xi_\ell(\omega), \tag{2}$$

where $\xi_\ell(\omega)$ are uncorrelated random variables and $E_\kappa(x)$ is the mean value of $\kappa(x,\omega)$. λ_ℓ and ϕ_ℓ are eigenvalues and eigenvectors of the following eigenproblem

$$T\phi_\ell = \lambda_\ell\phi_\ell, \quad \phi_\ell \in L^2(\mathcal{G}),\ \ell \in \mathbb{N}, \tag{3}$$

where operator T is defined like follows

$$T : L^2(\mathcal{G}) \to L^2(\mathcal{G}), \quad (T\phi)(x) := \int_{\mathcal{G}} \mathrm{cov}_\kappa(x,y)\phi(y)\mathrm{d}y,$$

where $\mathrm{cov}_\kappa(x,y)$ a given covariance function and $\mathcal{G}$ a computational domain. Throwing away all unimportant terms in (2), one obtains the truncated KLE, which is a sparse representation of the random field $\kappa(x,\omega)$. Each random variable ξ_ℓ can be, in its turn, approximated in a set of new independent

Gaussian random variables (polynomial chaos expansions (PCE) of Wiener [4, 21]), e.g.

$$\xi_\ell(\omega) = \sum_{\beta \in \mathcal{J}} \xi_\ell^{(\beta)} H_\beta(\boldsymbol{\theta}(\omega)), \tag{4}$$

where $\boldsymbol{\theta}(\omega) = (\theta_1(\omega), \theta_2(\omega), ...)$, $\xi_\ell^{(\beta)}$ are coefficients, H_β, $\beta \in \mathcal{J}$, is a Hermitian basis and $\mathcal{J} := \{\beta | \beta = (\beta_1, ..., \beta_j, ...),\ \beta_j \in \mathbb{N}_0\}$ a multi-index set (see Appendix or [14]). Computing the truncated PCE for each random variable in KLE, one can make representation of the random field (2) even more sparse.

Since Hermite polynomials are orthogonal, the coefficients $\xi_\ell^{(\beta)}$ in Eq. 4 can be computed by the following projection

$$\xi_\ell^{(\beta)} = \frac{1}{\beta!} \int_\Theta H_\beta(\boldsymbol{\theta}) \xi_\ell(\boldsymbol{\theta})\, \mathbb{P}(\mathrm{d}\boldsymbol{\theta}).$$

This multidimensional integral over Θ can be computed approximately, for example, on a sparse Gauss-Hermite grid

$$\xi_\ell^{(\beta)} = \frac{1}{\beta!} \sum_{i=1}^{n} H_\beta(\boldsymbol{\theta}_i) \xi_\ell(\boldsymbol{\theta}_i) w_i, \tag{5}$$

where weights w_i and points $\boldsymbol{\theta}_i$ are defined from sparse Gauss-Hermite integration rule.

The algorithms for construction of sparse Gauss-Hermite grids are well known (e.g., [8]). Three examples of two-dimensional sparse Gauss-Hermite grids

$$(\alpha_i, Ma_i), \quad i = 1..n,\ Z = \{13, 29, 137\}$$

are shown in Fig. 1.

After a finite element discretisation (see [7] for more details) the discrete eigenvalue problem (3) can be written in the following form

$$MCM\boldsymbol{\phi}_\ell = \lambda_\ell^h M\boldsymbol{\phi}_\ell, \quad C_{ij} = \mathrm{cov}_\kappa(x_i, y_j). \tag{6}$$

Here the mass matrix M is stored in a usual data sparse format and the dense matrix $C \in \mathbb{R}^{n \times n}$ (requires $\mathcal{O}(n^2)$ units of memory) is approximated in the sparse $\mathcal{H}$-matrix format [7] (requires only $\mathcal{O}(n \log n)$ units of memory) or in the Kronecker low-rank tensor format [6, 10]. To compute m largest eigenvalues ($m \ll n$) and corresponding eigenvectors we applied the Lanczos eigenvalue solver [9, 18].

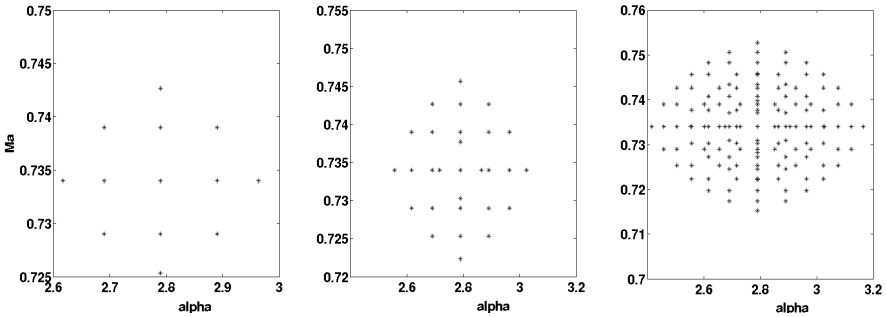

Figure 1: Sparse Gauss-Hermite grids for the uncertain angle of attack $\alpha^{'}$ and the Mach number $Ma^{'}$, $Z = \{13, 29, 137\}$.

3 Statistical modelling of uncertainties

We have implemented two different strategies to research simultaneous propagation of uncertainties in the angle of attack α and in the Mach number Ma on the solution. In the first strategy (Section 3.1) we assumed that the mean values and standard deviations for the random variables α and Ma are given. Then for each pair α_i and Ma_i of the corresponding sparse Gauss-Hermite grid we compute the deterministic solution via the TAU code (deterministic solver). After that the mean value, the variance as well as the density and cumulative distribution functions are computed. To validate the sparse Gauss-Hermite grid methods we compare the obtained results with the results of Monte Carlo simulations (reference solution). The second strategy (Section 3.2) assumes that the turbulence in the atmosphere randomly and simultaneously changes the velocity vector or, what is equivalent the Mach number, (Eq. 10) and the angle of attack (see Eq. 9 and Fig. 2). The turbulence in the atmosphere is modelled by two additionally axes-parallel velocity vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, which have Gaussian distribution.

3.1 Distribution functions of α and Ma are given

It is supposed that cumulative distribution functions of α and Ma are known, although in real-life applications it is not the case. As a start point we consider the uniform and the Gaussian distributions. For our further numerical experiments we choose the mean values and the standard deviations as in Table 5. The Reynolds number is $Re = 6.5e + 6$ and the computational

geometry is RAE-2822 airfoil.

3.2 Modelling of turbulence in the atmosphere

In this section we describe how uncertainties in the free-stream turbulence influence on the angle of attack α and on the Mach number (see Fig. 2). One should not mix this kind of turbulence with the turbulence in the boundary layer reasoned by friction. It is assumed that turbulence vortices in the atmosphere are comparable with the size of the airplane.

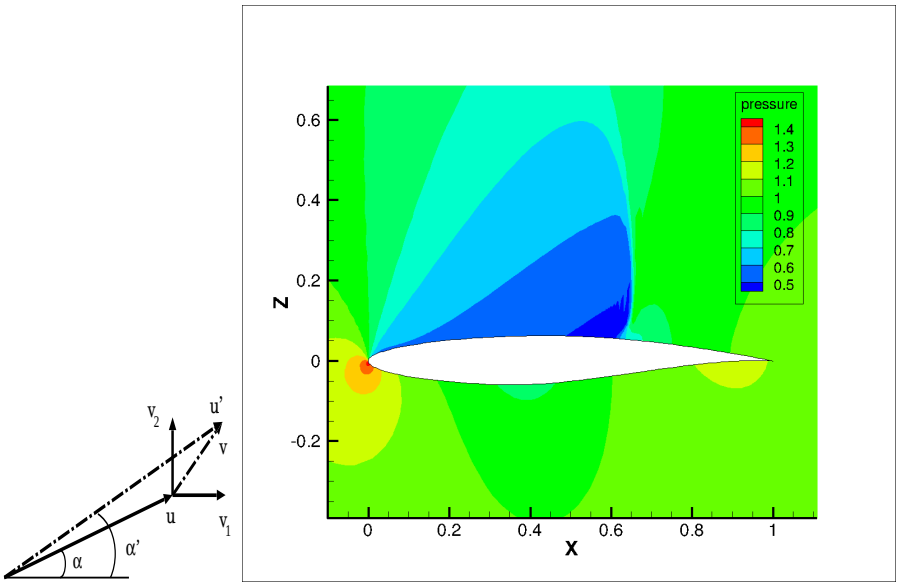

Figure 2: Two random vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ model free-stream turbulence, $\mathbf{u}$ and $\mathbf{u}^{'}$ old and new freestream velocities, α and $\alpha^{'}$ old and new angles of attack.

For further explanation, we remind definition of the mean turbulence intensity I, which can be computed as follows (in 3D-Space):

$$\mathtt{I} := \frac{\sigma}{u_\infty}, \quad \sigma = \sqrt{\frac{1}{3}(\overline{u_x}^2 + \overline{u_y}^2 + \overline{u_z}^2)}, \tag{7}$$

where u_∞ is the undisturbed freestream velocity beyond the boundary layer, $\overline{u_x}$, $\overline{u_y}$ and $\overline{u_z}$ are averaged variability of velocities in the directions x, y and

z correspondingly. This mean turbulence intensity is often used for characterising turbulence in a wind tunnel. By default, in the TAU code, the mean turbulence intensity is $I = 0.001$.
We model the turbulence in the atmosphere via two (for simplicity we consider 2D-Space) random vectors

$$\mathbf{v}_1 = \frac{\sigma\theta_1}{\sqrt{2}} \quad \text{and} \quad \mathbf{v}_2 = \frac{\sigma\theta_2}{\sqrt{2}},$$

where θ_1 and θ_2 two Gaussian random variables with zero mean and unit variance.
Denoting

$$\theta := \sqrt{\theta_1^2 + \theta_2^2},\ \mathbf{v} := \sqrt{\mathbf{v}_1^2 + \mathbf{v}_2^2},\ \beta := \text{arctg}\frac{\mathbf{v}_2}{\mathbf{v}_1} \text{ and } z := \frac{\mathtt{I}\theta}{\sqrt{2}}, \tag{8}$$

the new angle of attack and the new Mach number will be as follows (see Fig. 2 left)

$$\alpha^{'} = \text{arctg}\frac{\sin\alpha + z\sin\beta}{\cos\alpha - z\cos\beta}, \tag{9}$$

$$Ma^{'} = Ma\sqrt{1 + \frac{\mathtt{I}^2\theta^2}{2} - \sqrt{2}\mathtt{I}\theta\cos(\beta + \alpha)}. \tag{10}$$

Thus, alternatively to the way of modelling introduced in Sec.3.1, uncertainties in the angle of attack $\alpha^{'} = \alpha^{'}(\theta_1, \theta_2)$ and in the Mach number $Ma^{'} = Ma^{'}(\theta_1, \theta_2)$ are described via two standard normal variables θ_1 and θ_2. The pressure field in Fig. 2 and the shock position will be changed for the new angle of attack $\alpha^{'}$ and for the new Mach number $Ma^{'}$.

3.3 Uncertainties in geometry

Let us denote the airfoil geometry by $\mathcal{G}$ and surface of the airfoil by $\partial\mathcal{G}$. We model uncertainties in the airfoil geometry $\mathcal{G}$ by the usage of random field $\kappa(x, \omega)$:

$$\partial\mathcal{G}_\varepsilon(\omega) = \{x + \varepsilon\kappa(x, \omega)n(x) \,:\, x \in \partial\mathcal{G}\}, \tag{11}$$

where $n(x)$ is a normal vector in point x and ε a small parameter.
We assume that the covariance function $\text{cov}(p_1, p_2)$ for the random field $\kappa(x, \omega)$ is given. To generate Z realisations of RAE2822 airfoil with uncertain deformations (e.g., for MC simulations) we follow to the Algorithm below:

1. Compute sparse approximation of $\mathbf{C}_{ij} := \mathrm{cov}(p_i, p_j)$ for all grid points $i, j = 1..N$.

2. Compute m largest eigenvalues λ_i and corresponding eigenvectors $\phi_i(x)$, $i = 1..m$ of the eigenproblem (6).

3. Generate a random vector $\xi = (\xi_1(\omega), ..., \xi_m(\omega))$.

4. Generate a new realisation of the airfoil

$$\kappa(x, \omega) \approx \sum_{i=1}^{m} \sqrt{\lambda_i}\phi_i(x)\xi_i(\omega). \tag{12}$$

Sparse approximations of the dense matrix $\mathbf{C}$ are offered in [7, 6].
In Fig.3 one can see 21 realisations of RAE-2822 airfoil with uncertain deformations. One should note that ranges in x and y directions are different: $x \in [0, 1]$, $y \in [-0.08, 0.08]$. The used covariance function is of Gaussian type:

$$\mathrm{cov}(p_1, p_2) = a^2 \cdot \exp(-\rho^2), \tag{13}$$

where $a^2 = 10^{-5}$ is a parameter, $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2) \in \mathcal{G}$ two points, l_1, l_2 are correlation length scales, and

$$\rho(p_1, p_2) = \sqrt{|x_1 - x_2|^2/l_1^2 + |y_1 - y_2|^2/l_2^2}. \tag{14}$$

The quantify influence of uncertainties in the airfoil geometry $\gamma(x, \omega)$ on the solution we build the response surface as follows:
Algorithm: (**Building of response surface**)

1. Compute m largest eigenvalues and corresponding eigenpairs of the discrete eigenvalue problem Eq. 6.

2. Generate a sparse Gauss-Hermite grid in m-dimensional space with Z grid points.

3. For each grid point $\boldsymbol{\theta} = (\theta_1, ..., \theta_m)$ from item (2) compute new realisation of the airfoil like in Eq.(12).

4. For all of Z new airfoils solve the deterministic problem.

5. Using all Z solutions from item (4) and Hermite polynomials build the response surface.

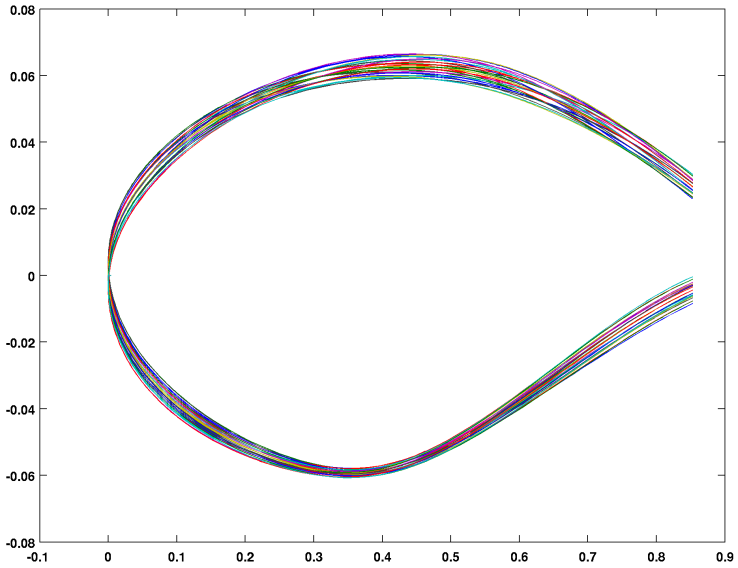

Figure 3: 21 realisations of RAE-2822 airfoil, computed for Gaussian covariance function $10^{-5} \cdot \exp(-(|x_1 - x_2|^2/l_1^2 + |y_1 - y_2|^2/l_2^2))$, $x \in [0, 1]$, $y \in [-0.08, 0.08]$, $l_1 = 0.3$ and $l_2 = 0.04$.

When the response surface is ready, we generate 10^6 random points $\boldsymbol{\theta}_i$, $i = 1..10^6$, and evaluate response surface in these points. Using the sample of size 10^6, evaluate statistical functionals of interest. If functionals of interest are, for instance, the lift CL and the drag CD, than the corresponding response surfaces will be $CL(\boldsymbol{\theta})$ and $CD(\boldsymbol{\theta})$.

4 Data compression

A large number of stochastic realisations of random fields requires a large amount of memory and powerful computational resources. To decrease memory requirements and the computing time we offer to use a low-rank approximation for all realisations of input and output random fields. For each new realisation only corresponding low-rank update will be computed (see, e.g. [1]). This can be practical when, e.g. the results of many thousands Monte Carlo simulations should be computed and stored.

Let $\mathbf{v}_i \in \mathbb{R}^n$ be the solution vector (already centred), where $i = 1..Z$ a number of stochastic realisations of the solution. Let us build from all these vectors the matrix $W = (\mathbf{v}_1, ..., \mathbf{v}_Z) \in \mathbb{R}^{n \times Z}$ and consider the factorization

$$W = AB^T, \quad \text{where} \quad A \in \mathbb{R}^{n \times k} \quad \text{and} \quad B \in \mathbb{R}^{Z \times k}. \tag{15}$$

Definition 1 *We say that matrix W is **a rank-k matrix** if the representation (15) is given. We denote the class of all rank-k matrices for which factors A and B^T in (15) exist by $\mathcal{R}(k, n, Z)$. If $W \in \mathcal{R}(k, n, Z)$ we say that W has a **low-rank representation**.*

The first aim is to compute a rank-k approximation $\tilde{W}$ of W, such that

$$\|W - \tilde{W}\| < \varepsilon, \quad k \ll \min\{n, Z\}.$$

The second aim is to compute an update for the approximation $\tilde{W}$ with a linear complexity for every new coming vector $\mathbf{v}_{Z+1}$. Below we present the algorithm which performs this.
To get the reduced singular value decomposition we omit all singular values, which are smaller than a given level ε or, alternative variant, we leave a fixed number of largest singular values. After truncation we speak about *reduced singular value decomposition (denoted by rSVD)* $\tilde{W} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$, where $\tilde{U} \in \mathbb{R}^{n \times k}$ contains the first k columns of U, $\tilde{V} \in \mathbb{R}^{Z \times k}$ contains the first k columns of V and $\tilde{\Sigma} \in \mathbb{R}^{k \times k}$ contains the k-biggest singular values of Σ. There is theorem (see more in [15] or [3]) which tells that matrix $\tilde{W}$ is the best approximation of W in the class of all rank-k matrices.
The computation of such basic statistics as the mean value, the variance, the exceedance probability can be done with a linear complexity. The following examples illustrate computation of the mean value and the variance.
Let $W = (\mathbf{v}_1, ..., \mathbf{v}_Z) \in \mathbb{R}^{n \times Z}$ and its rank-k representation $W = AB^T$, $A \in \mathbb{R}^{n \times k}$, $B^T \in \mathbb{R}^{k \times Z}$ be given. Denote the j-th row of matrix A by $\mathbf{a}_j \in \mathbb{R}^k$ and the i-th column of matrix B^T by $\mathbf{b}_i \in \mathbb{R}^k$.

1. One can compute the mean solution $\overline{\mathbf{v}} \in \mathbb{R}^n$ as follows

$$\overline{\mathbf{v}} = \frac{1}{Z} \sum_{i=1}^{Z} \mathbf{v}_i = \frac{1}{Z} \sum_{i=1}^{Z} A \cdot \mathbf{b}_i = A\overline{\mathbf{b}}, \tag{16}$$

   The computational complexity is $\mathcal{O}(k(Z+n))$, besides $\mathcal{O}(nZ)$) for usual dense data format.

2. One can compute the mean value of the solution in a grid point x_j as follows
$$\overline{\mathbf{v}}(x_j) = \frac{1}{Z}\sum_{i=1}^{Z}\mathbf{v}_i(x_j) = \frac{1}{Z}\sum_{i=1}^{Z}\mathbf{a}_j \cdot \mathbf{b}_i^T = \mathbf{a}_j\overline{\mathbf{b}}. \tag{17}$$
The computational complexity is $\mathcal{O}(kZ)$.

3. One can compute the variance of the solution $\mathrm{var}(\mathbf{v}) \in \mathbb{R}^n$ by the computing the covariance matrix and taking its diagonal. First, one computes the centred matrix $W_c := W - \overline{W}e^T$, where $\overline{W} = W \cdot e/Z$ and $e = (1, ..., 1)^T$. Computing W_c costs $\mathcal{O}(k^2(n + Z))$ (addition and truncation of rank-k matrices). By definition, the covariance matrix is $C = W_c W_c^T$. The reduced singular value decomposition of W_c is $W_c = U\Sigma V^T$, $U \in \mathbb{R}^{n\times k}$, $\Sigma \in \mathbb{R}^{k\times k}$ and $V \in \mathbb{R}^{Z\times k}$ can be computed via the QR algorithm (Section 4.1). Now, the covariance matrix can be written like
$$C = \frac{1}{Z-1}W_c W_c^T = \frac{1}{Z-1}U\Sigma V^T V\Sigma^T U^T = \frac{1}{Z-1}U\Sigma\Sigma^T U^T. \tag{18}$$
The variance of the solution vector (i.e. the diagonal of the covariance matrix in (18)) can be computed with the complexity $\mathcal{O}(k^2(Z + n))$.

4. One can compute the variance value $\mathrm{var}(\mathbf{v}(x_j))$ in a grid point x_j with a linear computational cost.

4.1 Low-rank update with linear complexity

Let $W = AB^T \in \mathbb{R}^{n\times Z}$ and matrices A and B be given. An rSVD $W = U\Sigma V^T$ can be computed efficiently in three steps (QR algorithm for computing the reduced SVD):

1. Compute (reduced) QR-factorization of $A = Q_A R_A$ and $B = Q_B R_B$, where $Q_A \in \mathbb{R}^{n\times k}$, $Q_B \in \mathbb{R}^{Z\times k}$, and upper triangular matrices $R_A, R_B \in \mathbb{R}^{k\times k}$.

2. Compute rSVD of $R_A R_B^T = U'\Sigma V'^T$.

3. Compute $U := Q_A U'$, $V := Q_A V'^T$.

QR-decomposition can be done faster if a part of matrix A (or B) is orthogonal (see [1]). The first and third steps need $\mathcal{O}((n+Z)k^2)$ operations and the second step needs $\mathcal{O}(k^3)$. The total complexity of rSVD is $\mathcal{O}((n+Z)k^2+k^3)$. Suppose we have already matrix $W = AB^T \in \mathbb{R}^{n\times Z}$ containing solution vectors. Suppose also that matrix $W^{'} \in \mathbb{R}^{n\times m}$ contains new m solution vectors. For the small matrix $W^{'}$, computing the factors C and D^T such that $W^{'} = CD^T$ is not expensive. Now our purpose is to compute with a linear complexity the new matrix $W_{\text{new}} := [W\, W'] \in \mathbb{R}^{n\times(Z+m)}$ in the rank-k format. To do this, we build two concatenated matrices $A_{\text{new}} := [A\, C] \in \mathbb{R}^{n\times 2k}$ and $B^T_{\text{new}} = \text{blockdiag}[B^T\, D^T] \in \mathbb{R}^{2k\times(Z+m)}$. Note that the difficulty now is that matrices A_{new} and B_{new} have rank $2k$. To truncate the rank from $2k$ to k we use the QR-algorithm above:

$$W_{\text{new}} = U\Sigma V^T = U(V\Sigma^T)^T = A_{\text{new}}B^T_{\text{new}},$$

where $A_{\text{new}} \in \mathbb{R}^{n\times k}$ and $B^T_{\text{new}} \in \mathbb{R}^{k\times(Z+m)}$. Thus, the rank k approximation of the new matrix W_{new} is done with a linear complexity $\mathcal{O}((n+Z)k^2+k^3)$.

5 Numerics

Further numerical results are obtained in the MUNA (management and minimization of uncertainties in numerical aerodynamics) project [17]. We demonstrate propagation of uncertainties in the angle of attack, the Mach number and the airfoil geometry on the solution (the lift, drag, lift and skin friction coefficients). As an example we consider two-dimensional RAE-2822 airfoil. The deterministic solver is the TAU code with k-w turbulence model is developed in DLR [19]. We assume that α and Ma are Gaussian with means $\overline{\alpha} = 2.79$, $\overline{Ma} = 0.734$ and the standard deviations $\sigma(\alpha) = 0.1$ and $\sigma(Ma) = 0.005$ (Table 1, on the left). To quantify uncertainties we used the collocation method computed in nodes of sparse Gauss-Hermite two-dimensional grid (with $Z = 5$ grid points). The Hermite polynomials are of order 1 with two random variables (see (4)). The last column in Tables 1 on the left and on the right shows the measure of uncertainty σ/mean. It shows that 3.6% and 0.7% of uncertainties in α and in Ma correspondingly result in 2.1% and 15.1% (Table 1, on the right) of uncertainties in the lift CL and drag CD.

	mean	st. dev. σ	σ/mean
α	2.79	0.1	0.036
Ma	0.734	0.005	0.007

$\Longrightarrow$

	mean	st. dev. σ	σ/mean
CL	0.853	0.018	0.021
CD	0.0206	0.0031	0.151

Table 1: Uncertainties in the input parameters (α and Ma) and in the solution (CL and CD). PCE of order 1 and a sparse Gauss-Hermite grid with 5 points.

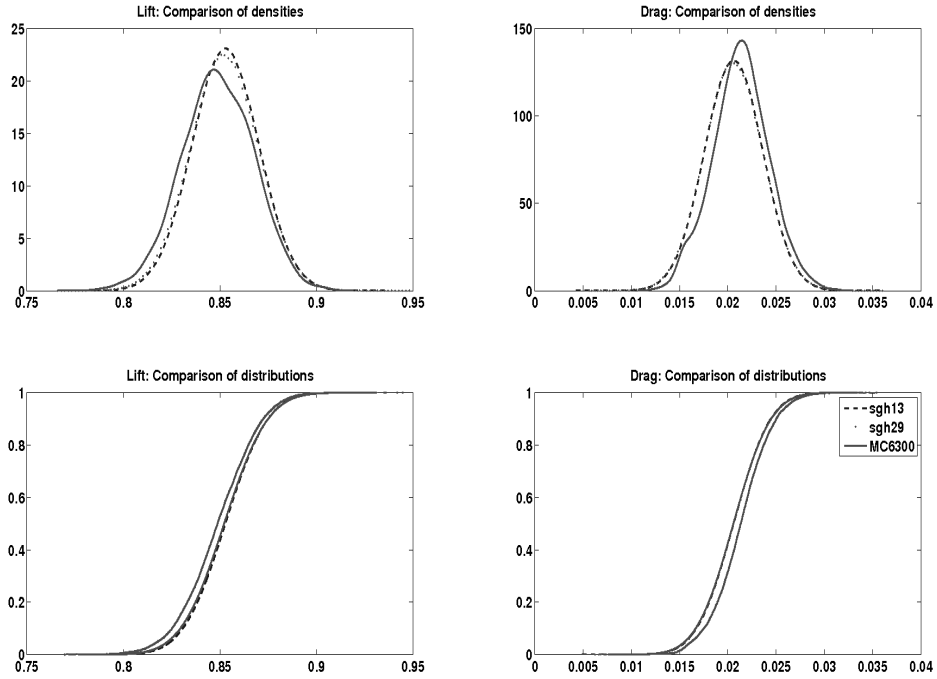

Figure 4: Density functions (first row), cumulative distribution functions (second row) of CL (left) and CD (right). PCE is of order 1 with two random variables. Three graphics computed with 6360 MC simulations, 13 and 29 collocation points.

In Fig.4 we compare the cumulative distribution and density functions for the lift and drag, obtained via the response surface (PCE of order 1) and via 6360 Monte Carlo simulations. To build the response surface we used, first, the sparse Gauss-Hermite grid with 13 nodes, and than with 29 nodes. On each response surface 10^6 MC evaluations were performed. Thus, one

can see that response surfaces (13 or 29 deterministic evaluations) produce similar to MC method (6360 simulations) results. But, at the same time we can not say which result is more precise. The exact solution is unknown and 6360 MC simulations are too few for the reference solution.
The graphics in Fig. 5 demonstrate 3σ error bars, σ the standard deviation, for the pressure *cp* and absolute skin friction *cf* coefficients in each surface point of the RAE2822 airfoil. The data are obtained from 645 realisation of the solution. One can see that the largest error occur at the shock ($x \approx 0.6$). A possible explanation is that the shock position is expected to slightly change with varying parameters α and *Ma*.

In Figure 6 one can see 5% and 95% quantiles for the pressure *cp* and the

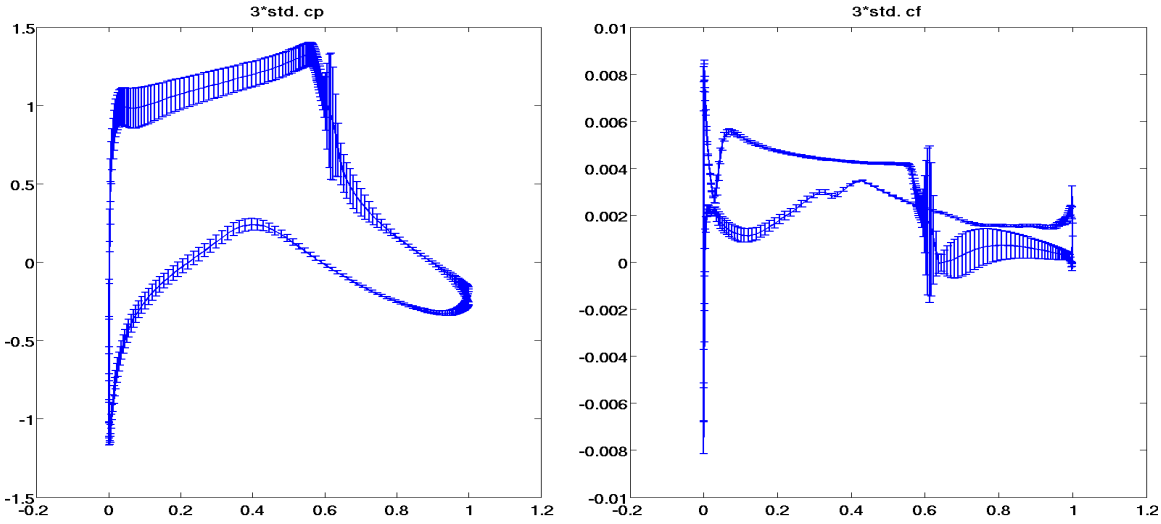

Figure 5: 3σ error bars in each point of RAE2822 airfoil for the *cp* and *cf*.

skin friction *cf* coefficients.
To decrease memory requirements we write all $Z = 645$ realisations of the solution as matrices $\in \mathbb{R}^{512\times 645}$ and compute their rank-k approximations.

In Table 2 one can see dependence of the relative error (in the spectral norm) on the rank k. Additionally, one can also see much smaller memory requirement (dense matrix format costs 2.6MB). In the two last rows we compare computing time needed for SVD-update Algorithm described in Section 4.1 with the standard SVD of the global matrix $\in \mathbb{R}^{512\times 645}$. One can

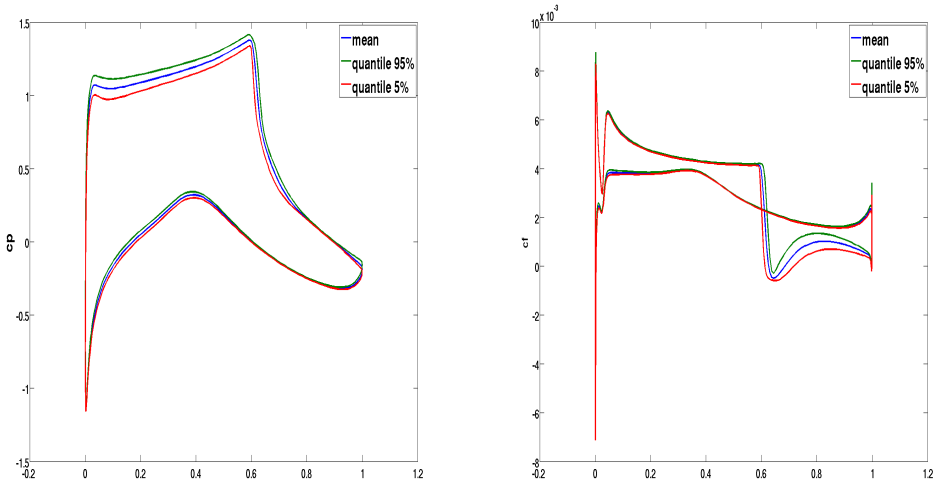

Figure 6: 5% and 95% quantiles for the pressure *cp* and the absolute skin friction *cf* coefficients

see that SVD-update Algorithm performs faster.

rank k	2	5	10	20
$\|D - \tilde{D}_k\|_2/\|D\|_2$	6.6e-1	4.1e-2	3.5e-3	3.5e-4
$\|P - \tilde{P}_k\|_2/\|P\|_2$	6.9e-1	8.4e-2	8.2e-3	7.2e-4
$\|CP - \tilde{CP}_k\|_2/\|CP\|_2$	6.0e-3	5.3e-4	3.2e-5	2.4e-6
$\|CF - \tilde{CF}_k\|_2/\|CF\|_2$	9.0e-3	7.7e-4	4.6e-5	3.5e-6
memory, kB	18	46	92	185
Update time, sec	0.58	0.60	0.62	0.68
usual SVD time, sec	0.55	0.63	2.6	3.8

Table 2: Accuracy, computing time and memory requirements of the rank-k approximation of the solution matrices D = [density], P = [pressure], CP = [cp]; CF = [cf] $\in \mathbb{R}^{512\times 645}$.

Fig. 7 demonstrates decay of 100 largest eigenvalues of four matrices, corresponding to the pressure, density, pressure coefficient *cp* and absolute skin friction *cf*. Each matrix belongs to the space $\mathbb{R}^{512\times 645}$.

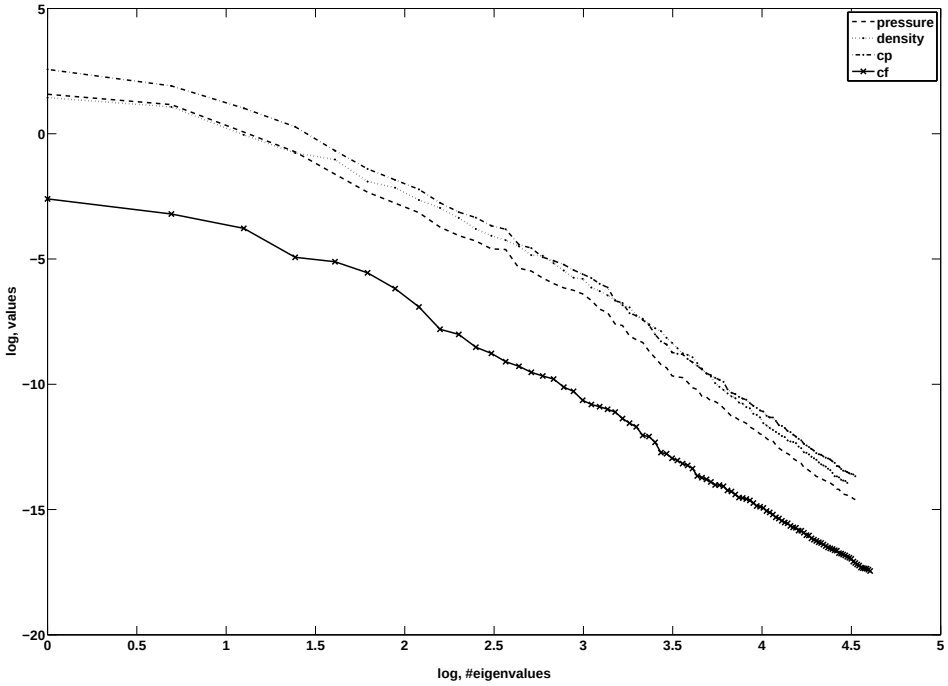

Figure 7: Decay (in log-scales) of 100 largest eigenvalues of the matrices constructed from 645 solutions (pressure, density, *cf*, *cp*) on the surface of RAE-2822 airfoil.

In Table 3 one can see dependence of the relative error (in the Frobenious norm) on the rank k. Seven solution matrices contain pressure, density, turbulence kinetic energy (tke), turbulence omega (to), eddy viscosity (ev), x-velocity (xv), z-velocity (zv) in the whole computational domain with 260000 dofs. Additionally, one can also see much smaller memory requirement (dense matrix format costs 1.25GB). In Table 4 one can see corresponding computing times: time required for the SVD-update Algorithm described in Section 4.1 and the time required for the standard SVD of the global matrix $\in$

$\mathbb{R}^{260000\times600}$. A possible explanation for the large computing time for the standard SVD is the lack of memory and expansive swapping of data.

rank k	pressure	density	tke	to	ev	xv	zv	memory, MB
10	1.9e-2	1.9e-2	4.0e-3	1.4e-3	1.4e-3	1.1e-2	1.3e-2	21
20	1.4e-2	1.3e-2	5.9e-3	3.3e-4	4.1e-4	9.7e-3	1.1e-2	42
50	5.3e-3	5.1e-3	1.5e-4	9.1e-5	7.7e-5	3.4e-3	4.8e-3	104

Table 3: Relative errors and memory requirements of rank-k approximations of the solution matrices $\in \mathbb{R}^{260000\times600}$. Memory required for the storage of each matrix in the dense matrix format is 1.25 GB.

rank k	Update time, sec.	SVD time, sec.
10	107	1537
20	150	2084
50	228	8236

Table 4: Computing times (for Table 3) of rank-k approximations of the solution matrices $\in \mathbb{R}^{260000\times600}$.

Figure 8 explains why it was possible to achieve so high data compression factor in Table 3. One can see that only a small part of the whole computational domain contains something "interesting" (schock, separation, turbulent eddies etc). This part may require a high approximation rank, whereas the rest of the domain can be approximated by a low-rank.

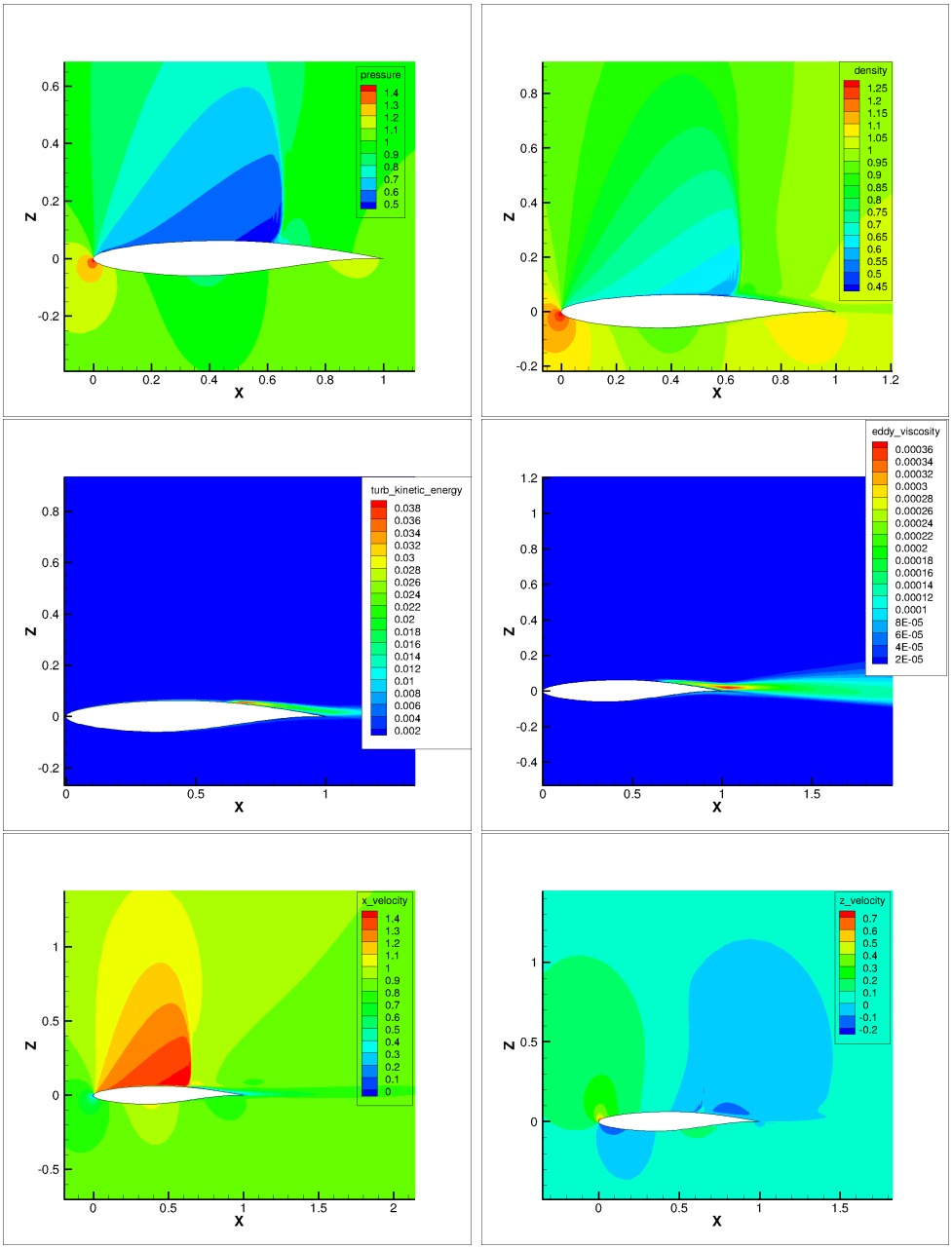

Figure 8: An example of realisations of pressure, density (the first row), turbulence kinetic energy, eddy viscosity (the second row), velositsy in x and z directions (the third row).

5.1 α and *Ma* have Gaussian/uniform distribution

Our assumption is that the input parameters α and Ma have Gaussian distribution with mean values and standard deviations as in Table 5. Table

	mean	st. deviation, σ	σ/mean
Angle of attack, α	2.790	0.1	0.036
Mach number, Ma	0.734	0.005	0.007

Table 5: Mean values and standard deviations

6 demonstrates application of sparse Gauss-Hermite two-dimensional grids with $Z = \{5, 13, 29\}$ grid points. The Hermite polynomials (see Eq. 4) are of order 1 with two random variables. In the last column we compute the measure of uncertainty σ/mean. For instance, for $Z = 5$ it shows that 3.6% and 0.7% (Table 5) of uncertainties in α and in Ma correspondingly result in 2.1% and 15.1% of uncertainties in the lift and drag coefficients (Table 6, $Z = 5$). These three grids ($Z = 5, 13, 29$) show very similar results in the mean value and in the standard deviation.

Z		mean	st. dev. σ	σ/mean
5	CL	0.8530	0.0180	0.021
	CD	0.0206	0.0031	0.151
13	CL	0.8530	0.0174	0.020
	CD	0.0206	0.0030	0.146
29	CL	0.8520	0.0180	0.021
	CD	0.0206	0.0031	0.151
MC 1500	CL	0.8525	0.0172	0.020
	CD	0.0206	0.0030	0.146

Table 6: Uncertainties obtained on sparse Gauss-Hermite grids with 5 ,13, 29 points and with 1500 MC simulations.

At the same time the results obtained via 1500 MC simulations are very similar to the results computed on all three sparse Gauss-Hermite grids.
Table 7 demonstrates statistics obtained for the case when random variables α and Ma have uniform distributions. Comparing Table 7 with Table 6 one

can see that, in the case of uniform distribution of uncertain parameters, the uncertainties in the lift and drag coefficients are smaller. Namely, 1.2% and 8.8% (for CL and CD) against 2% and 14.6% in the case of the Gaussian distribution. But, uncertainties in the input parameters α and Ma, in the case of the uniform distribution, are also smaller: 2.1% and 0.4% against 3.5% and 0.7%.

	mean	st. dev. σ	σ/mean
α	2.787	0.058	0.021
Ma	0.734	0.003	0.004

$\Longrightarrow$

	mean	st. dev. σ	σ/mean
CL	0.853	0.0104	0.012
CD	0.0205	0.0018	0.088

Table 7: Uncertainties in the input parameters (α and Ma) and in the solution (CL and CD). Estimations are obtained from 3800 MC simulations, where α and Ma have uniform distributions.

5.2 $\alpha(\theta_1, \theta_2)$, $\boldsymbol{Ma}(\theta_1, \theta_2)$, where θ_1, θ_2 have Gaussian distributions

In this section we illustrate numerical results for the model described in Section 3.2.
Table 8 shows statistics (the mean value and the standard deviation), computed on sparse Gauss-Hermite grids with $Z = 137$ grid points.
Table 9 compares uncertainties computed on sparse Gauss-Hermite grids with $Z = \{137, 381, 645\}$ nodes with the uncertainties computed by the MC method (17000 simulations). All three grids and MC forecast very similar uncertainties σ/mean in the drag coefficient CD and in the lift coefficient CL.

	mean	st. dev. σ	σ/mean
α	2.8	0.2	0.071
Ma	0.73	0.0026	0.004

$\Longrightarrow$

	mean	st. dev. σ	σ/mean
CL	0.85	0.0373	0.044
CD	0.01871	0.00305	0.163

Table 8: Uncertainties in the input parameters (α and Ma) and in the solution (CL and CD). Statistics obtained on a sparse Gauss-Hermite grid with 137 points.

Z	137	381	645	MC, 17000
$\frac{\sigma_{CL}}{\overline{CL}}$	0.044	0.042	0.042	0.045
$\frac{\sigma_{CD}}{\overline{CD}}$	0.163	0.159	0.16	0.159
$\frac{\lvert\overline{CL}-CL_0\rvert}{\overline{CL}}$	7.6e-4	1.3e-3	1.6e-3	4.2e-4
$\frac{\lvert\overline{CD}-CD_0\rvert}{\overline{CD}}$	1.66e-2	1.46e-2	1.4e-2	2.1e-2

Table 9: Comparison of results obtained by a sparse Gauss-Hermite grid (Z grid points) with 17000 MC simulations.

Table 10 compares relative errors computed on different sparse Gauss-Hermite grids. One can see that the errors are very small, thus sparse Gauss-Hermite grid with Z points can be successfully used to compute the mean values $\overline{CL}$ and $\overline{CD}$.

Z	137	381	645
$\frac{\lvert\overline{CL}_n-\overline{CL}_{MC}\rvert}{\overline{CL}_{MC}} \cdot 100\%$	0.1%	0.1%	0.1%
$\frac{\lvert\overline{CD}_n-\overline{CD}_{MC}\rvert}{\overline{CD}_{MC}} \cdot 100\%$	0.4%	0.6%	0.7%

Table 10: Comparison of mean values obtained by MC simulations and by sparse Gauss-Hermite grid with Z grid points.

5.3 Uncertainties in the geometry

We follow the Algorithm described in Section 3.3. The number of KLE terms is $m = 3$. The covariance function is of Gaussian type. The stochastic dimension is 3 and number of sparse Gauss-Hermite points is 25. Table 11 demonstrate the computed statistics. Surprisingly small are uncertainties in the CL and CD — 0.58% and 0.65% correspondingly. A possible explanation can be a small uncertain perturbations in the airfoil geometry.

6 Conclusion

In this work we researched how uncertainties in the input parameters (the angle of attack α and the Mach number Ma) and in the airfoil geometry

	mean	st. dev. σ	σ/mean
CL	0.8552	0.0049	0.0058
CD	0.0183	0.00012	0.0065

Table 11: Propagation of uncertainties in the airfoil geometry. Covariance function is of Gaussian type, PCE of order 1 with 3 random variables. Sparse Gauss-Hermite grid contains 25 points.

propagate to the solution (lift, drag, pressure and absolute skin friction coefficients). Uncertainties in the Mach number and in the angle of attack weakly affect the lift coefficient ($1\% - 3\%$) and strongly affect the drag coefficient (around 14%). Uncertainties in the geometry influence both the lift and drag coefficients weakly (less that 1%), but changes in the geometry were also very small. Results obtained via response surface, which is built on a sparse Gauss-Hermite grid are comparable with Monte Carlo results, but require much less deterministic evaluations (and as a sequence - smaller computing time).
From Tables 9 and 10 one can see that the results computed on a sparse Gauss-Hermite grid do not converge. We note that to get reliable results with Monte Carlo methods one should perform 10^5-10^7 simulations, but it is impossible to do in a reasonable time (1 simulation with the TAU code requires at least 20 minutes). We performed 17000 MC simulations and this is not enough for an accurate reference solution.
To make statistical computations more efficient (linear complexity and linear storage besides quadratic or even cubic) an additional research was devoted to a low-rank data format for storage of realisations of the solution. This low-rank format allows us to compute all important statistics with a linear complexity and drastically reduces memory requirements.

Acknowledgement. It is acknowledged that this research has been conducted within the project MUNA under the framework of the German Luftfahrtforschungsprogramm funded by the Ministry of Economics (BMWA). The authors would like also to thank Elmar Zandler for his matlab package "Stochastic Galerkin library" [23].

References

[1] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra Appl.*, 415(1):20–30, 2006.

[2] Qian-Yong Chen, David Gottlieb, and Jan S. Hesthaven. Uncertainty analysis for the steady-state flows in a dual throat nozzle. *Journal of Computational Physics*, 204(1):378 – 398, 2005.

[3] G. H. Golub and Ch. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[4] T. Hida, Hui-Hsiung Kuo, J. Potthoff, and L. Streit. *White noise - An infinite-dimensional calculus*, volume 253 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1993.

[5] Serhat Hosder, R. Walters, and R. Perez. A non-intrusive polynomial chaos method for uncertainty propagation in cfd simulations. *44th AIAA Aerospace Sciences Meeting and Exhibit, January 2006, Reno, Nevada*, (AIAA 2006-891):209–236, 2006.

[6] B. N. Khoromskij and A. Litvinenko. Data sparse computation of the Karhunen-Loève expansion. *Numerical Analysis and Applied Mathematics: International Conference on Numerical Analysis and Applied Mathematics, AIP Conf. Proc.*, 1048(1):311–314, 2008.

[7] B. N. Khoromskij, A. Litvinenko, and H. G. Matthies. Application of hierarchical matrices for computing the Karhunen-Loève expansion. *Computing*, 84(1-2):49–67, 2009.

[8] A. Klimke. Sparse grid interpolation toolbox: www.ians.uni-stuttgart.de/spinterp. 2008.

[9] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Research Nat. Bur. Standards*, 45:255–282, 1950.

[10] A. Litvinenko and H. G. Matthies. Sparse data representation of random fields. *Proceedings in Applied Mathematics and Mechanics, PAMM Vol. 9, Wiley-InterScience*, pages 587 – 588, 2009.

[11] A. Litvinenko and H. G. Matthies. Sparse data formats and efficient numerical methods for uncertainties quantification in numerical aerodynamics. *IV European Congress on Computational Mechanics (ECCM IV): Solids, Structures and Coupled Problems in Engineering*, 2010.

[12] M. Loève. *Probability theory I. Graduate Texts in Mathematics, Vol. 45, 46.* Springer-Verlag, New York, fourth edition, 1977.

[13] L. Mathelin, M. Y. Hussaini, and T. A. Zang. Stochastic approaches to uncertainty quantification in CFD simulations. *Numer. Algorithms*, 38(1-3):209–236, 2005.

[14] H. G. Matthies. Uncertainty quantification with stochastic finite elements. 2007. Part 1. Fundamentals. Encyclopedia of Computational Mechanics.

[15] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math. Oxford Ser. (2)*, 11:50–59, 1960.

[16] Habib N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. In *Annual review of fluid mechanics. Vol. 41*, volume 41 of *Annu. Rev. Fluid Mech.*, pages 35–52. Annual Reviews, Palo Alto, CA, 2009.

[17] Project:. Management and minimization of uncertainties in aerodynamics. *http://www.dlr.de/as/muna*, 2007-2010.

[18] Y. Saad. *Numerical methods for large eigenvalue problems.* Algorithms and Architectures for Advanced Scientific Computing. Manchester University Press, Manchester, 1992.

[19] D. Schwamborn, T. Gerhold, and R. Heinrich. The dlr tau-code: recent applications in research and industry. *European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006, P. Wesseling, E. Oñate, J. Périaux (Eds) TU Delft, The Netherlands, 2006*, 2006.

[20] Xiaoliang Wan and George Em Karniadakis. Long-term behavior of polynomial chaos in stochastic flow simulations. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5582 – 5596, 2006. John H. Argyris Memorial Issue. Part II.

[21] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60:897–936, 1938.

[22] J. A. S. Witteveen, A. Loeven, and H. Bijl. An adaptive stochastic finite elements approach based on newton-cotes quadrature in simplex elements. *Computers & Fluids*, 38(6):1270 – 1288, 2009.

[23] E. Zander. A malab/octave toolbox for stochastic galerkin methods: http://ezander.github.com/sglib/. 2008.

Technische Universität Braunschweig
Informatik-Berichte ab Nr. 2007-02

2007-02	J. Rang	Design of DIRK schemes for solving the Navier-Stokes-equations
2007-03	B. Bügling, M. Krosche	Coupling the CTL and MATLAB
2007-04	C. Knieke, M. Huhn	Executable Requirements Specification: An Extension for UML 2 Activity Diagrams
2008-01	T. Klein, B. Rumpe (Hrsg.)	Workshop Modellbasierte Entwicklung von eingebetteten Fahrzeugfunktionen, Tagungsband
2008-02	H. Giese, M. Huhn, U. Nickel, B. Schätz (Hrsg.)	Tagungsband des Dagstuhl-Workshopss MBEES: Modellbasierte Entwicklung eingebetteter Systeme IV
2008-03	R. van Glabbeek, U. Goltz, J.-W. Schicke	Symmetric and Asymmetric Asynchronous Interaction
2008-04	R. van Glabbeek, U. Goltz, J.-W. Schicke	On Synchronous and Asynchronous Interaction in Distributed Systems
2008-05	M. V. Cengarle, H. Grönniger B. Rumpe	System Model Semantics of Class Diagrams
2008-06	M. Broy, M. V. Cengarle, H. Grönniger B. Rumpe	Modular Description of a Comprehensive Semantics Model for the UML (Version 2.0)
2008-07	C. Basarke, C. Berger, K. Berger, K. Cornelsen, M. Doering J. Effertz, T. Form, T. Gülke, F. Graefe, P. Hecker, K. Homeier F. Klose, C. Lipski, M. Magnor, J. Morgenroth, T. Nothdurft, S. Ohl, F. Rauskolb, B. Rumpe, W. Schumacher, J. Wille, L. Wolf	2007 DARPA Urban Challenge Team CarOLO - Technical Paper
2008-08	B. Rosic	A Review of the Computational Stochastic Elastoplasticity
2008-09	B. N. Khoromskij, A. Litvinenko, H. G. Matthies	Application of Hierarchical Matrices for Computing the Karhunen-Loeve Expansion
2008-10	M. V. Cengarle, H. Grönniger B. Rumpe	System Model Semantics of Statecharts
2009-01	H. Giese, M. Huhn, U. Nickel, B. Schätz (Herausgeber)	Tagungsband des Dagstuhl-Workshops MBEES: Modellbasierte Entwicklung eingebetteter Systeme V
2009-02	D. Jürgens	Survey on Software Engineering for Scientific Applications: Reuseable Software, Grid Computing and Application
2009-03	O. Pajonk	Overview of System Identification with Focus on Inverse Modeling
2009-04	B. Sun, M. Lochau, P. Huhn, U. Goltz	Parameter Optimization of an Engine Control Unit using Genetic Algorithms
2009-05	A. Rausch, U. Goltz, G. Engels, M. Goedicke, R. Reussner	LaZuSo 2009: 1. Workshop fr langlebige und zukunftsfähige Softwaresysteme 2009
2009-06	T. Müller, M. Lochau, S. Detering, F. Saust, H. Garbers, L. Märtin, T. Form, U. Goltz	Umsetzung eines modellbasierten durchgängigen Enwicklungsprozesses für AUTOSAR-Systeme mit integrierter Qualitätssicherung
2009-07	M. Huhn, C. Knieke	Semantic Foundation and Validation of Live Activity Diagrams
2010-01	A. Litvinenko and H. G. Matthies	Sparse data formats and efficient numerical methods for uncertainties quantification in numerical aerodynamics